




Article

Secure Learning Management System Based on User Behavior

Alin Zamfiroiu ^{1,2,*}, Diana Constantinescu ¹, Mădălina Zurini ¹ and Cristian Toma ¹

¹ Department of Economic Informatics and Cybernetics, Faculty of Cybernetics, Statistics and Economic Informatics, Bucharest University of Economic Studies, 010552 Bucharest, Romania; dianacristinacons@gmail.com (D.C.); madalina.zurini@csie.ase.ro (M.Z.); cristian.toma@ie.ase.ro (C.T.)

² National Institute for Research & Development in Informatics, 011455 Bucharest, Romania

* Correspondence: alin.zamfiroiu@csie.ase.ro

Received: 9 October 2020; Accepted: 28 October 2020; Published: 31 October 2020



Abstract: The COVID-19 outbreak is an international problem and has affected people and students all over the world. When lockdowns were imposed internationally, learning management systems began to be used more than in the previous period. These systems have been used also for traditional forms of learning and not only for online learning. This pandemic has highlighted the need for online learning systems in the educational environment, but it is very important for these systems to be secure and to verify the authenticity of the students when they access a course or evaluation questions. In this period, everything is moving towards the digital world, with students that are connected from a distance to online systems. All activities in the educational environment will soon be performed digitally on learning management systems, which includes also the evaluation process of the students. In this paper, we propose a secure learning management system that uses the student's behavior to identify if they are an authentic student or not. This system can support the teacher's activities in the learning process and verify the authenticity of the students logged on to the system. This paper is aimed at learning management system developers, who can use the proposed algorithms in their developed platforms, and also at teachers, who should understand the importance of the identification of students on these platforms.

Keywords: e-learning; behavior; platform; algorithm; biometric

1. Introduction

Biometrics is the mechanism of recognizing a person through their distinctive features. This mechanism refers to physical and behavioral characteristics. Biometrics based on physical characteristics measures a static feature that does not change over time, such as fingerprints, face recognition, or hand, palm, or iris geometry. When it is based on behavioral characteristics, biometrics measures a person's features through the way they act, such as voice recognition, signature verification, keyboard dynamics, or mouse usage dynamics [1–3].

The advantage of biometrics based on physical traits is the high accuracy it presents in comparison to behavioral biometrics. However, additional equipment is required for the implementation of physical biometrics, leading to certain limitations, such as the high cost of purchasing the devices. Behavioral biometrics is a method often used for authentication, but it produces insufficient accuracy as the behavior is unstable and can change over time.

Identification and verification are the goals of both biometric techniques (physical and behavioral) for user recognition. The verification stage determines if a person is who it claim to be. Biometric authentication based on physical features is very easy to handle and at the same time difficult to counterfeit, compared to other traditional methods such as username and password.

User authentication or identification based on biometric methods is carried out in four stages [4,5]:

- Collection: This is used in the registration, identification and verification phases, when the system collects physical or behavioral data about the users who are interacting with the specific system;
- Extraction: The model or the distinct characteristics are extracted, and based on these, the profile of each user is created;
- Comparison: The created profile is compared to the one stored in the database;
- Match/mismatch: The system decides after comparison whether the features of the two profiles are similar or not.

These steps are shown in Figure 1.

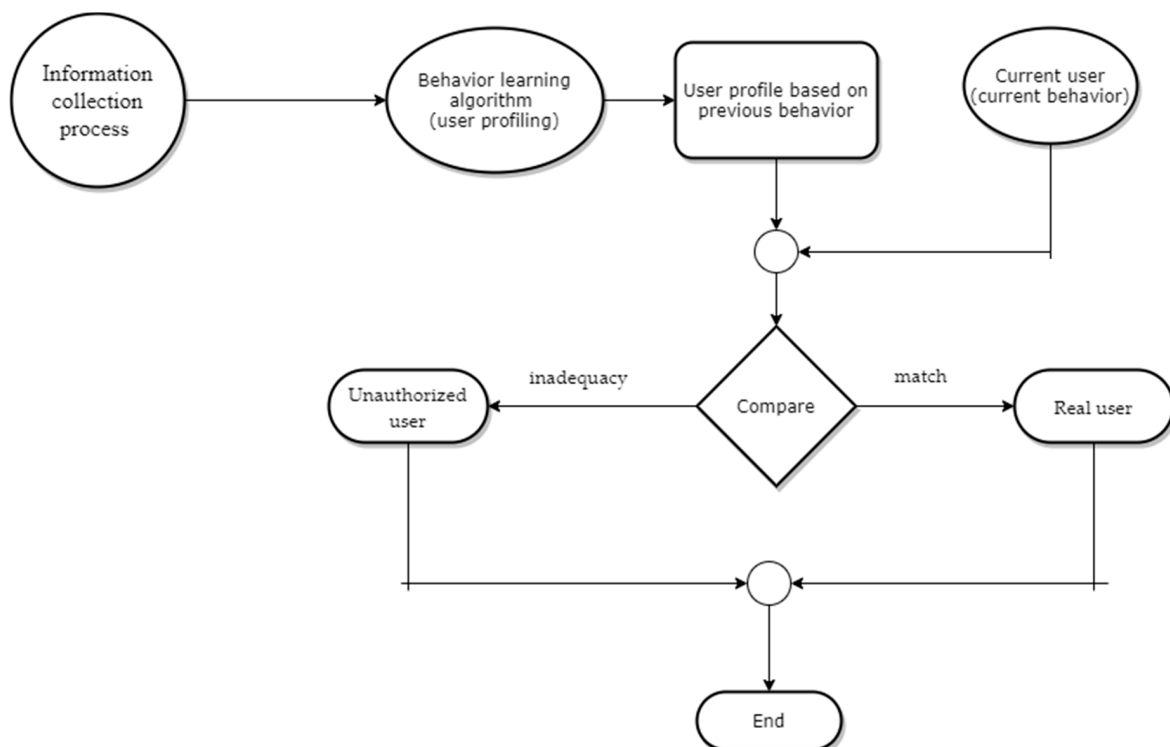


Figure 1. Identification of the user based on previous behavior.

Keyboard writing, as a biometric method of authentication, has been frequently used in recent years. By analyzing each user's way of writing, it was concluded that the typing rhythm often represents an eloquent model for the extraction and determination of the behavioral characteristics of each person. The input data consist of a flow of keyboard events (each key pressed represents an event) and the time taken by each of them. Each event is generated either by pressing a key or by releasing it. Most authentication techniques use either the time (in milliseconds) between two consecutive events, or the time it takes to type a group of characters (known as "digraph time") or an entire sequence.

We propose the following aims for this research:

- A better identification and understanding of the characteristics and requirements of each type of scenario (authentication as well as the duration of the working session) for continuous verification of identity;
- Implementation of algorithms that represent the writing behavior of student users and inherently reflect their profile;
- Development of a model for identifying users based on static behavior (the way a user types their username and password) to successfully authenticate;

- Creation of a technique that identifies users based on dynamic behavior in order to successfully create profiles and establish the authenticity of users after a work session.

This research is necessary because, due to the current situation, more schools and universities are adopting online education. In this way, teachers should be sure that the students are the correct students in the examination process. An approach that is being increasingly adopted by teachers is to connect online and via webcam to exactly see the student, but this approach is time-consuming. In this paper, we propose an option for automatic recognition of the students on the platform. We have analyzed some previous research studies in this field, and based on their research, we propose a new algorithm that uses more techniques from the previous researches.

In this paper, we propose a methodology and an algorithm that will improve the recognition of the user based on their behavior on the platform. Section 2 presents different approaches for the authentication of users on platforms, and in Section 3 we present the methods and algorithms that we have used in our developed platform. In our implemented platform, we have used three algorithms to improve the process of identification of the correct user who is authenticated in the platform. All these three algorithms are presented in Section 3. In Section 4, we present the implemented platform and, after that, the obtained results in the process of validation of the implemented algorithm on our platform.

2. Related Works

There are two types of keystroke analysis, continuous (dynamic) keystroke and static keystroke [6,7]. Static keystroke analysis involves extracting the characteristics of each user from the same predefined texts. Dynamic keystroke analysis means continuous monitoring of typing throughout the user's session, as soon as the authentication has been successfully completed. In this case, the system analyzes any type of text entered by the user. The two typing methods (static and dynamic) are detailed below, and the main research works in this area are presented [8–10].

Static authentication refers to login users using simple methods such as username and password. Static authentication behavior refers to how the user types the username and password. This method is used for further verification and to overcome some limitations of traditional methods of connection [11]. Keyboard dynamics and mouse dynamics are the main examples of static authentication based on user behavior [12,13].

These methods measure two indicators: False Reject Rate (FRR), when the system incorrectly rejects an access attempt by an authorized user, and False Acceptance Rate (FAR), when the system incorrectly accepts an access attempt by an unauthorized user. Table 1 describes some techniques that have been developed over time for static keystroke dynamics [14].

Table 1. Static methods for authentication.

Research	Number of Users	FAR (%)	FRR (%)	Sample Content	Method
[15]	6	0.00	0.00	6000 characters	Manual
[16]	17	12.00	6.00	1400 characters for training and 300 characters for testing	Statistical
[17]	15	0.00	0.00	Username 225 times	Neural network
[18]	154	0.00	0.14	683 characters	Nearest neighbor

In [15], keystroke dynamics were used for the first time as an authentication method. The authors did an experiment with six users, each user was asked to type two texts containing three paragraphs of varying length, and the time interval between collecting the two samples was four months. The five most common groups of letters were: in, io, no, on, and ul. They compared the time period between

the two sessions to see if the average was the same. An impediment to this experiment was that the data collected were not sufficient to have reliable results.

In [16], a sample of seventeen people was used to write two texts. The first text was approximately 1400 characters long and was used for preparation, and the second text was 300 characters long and was intended for testing. They calculated the time of the first digraphs for each word. The classifier was based on statistical methods by setting the condition that each digraph must fall within 0.5 standard deviations of its mean to be considered valid.

In [17], the entire word typing duration and the duration between two consecutive characters were used to distinguish between a valid user and an unauthorized user. The experiment used fifteen users who had to type their username 225 times a day for eight weeks. Neural networks helped to classify user samples, and the results showed that both FRR and FAR were zero.

In [18], a text of 683 characters was used for 154 participants, and it was considered that typing errors and the natural typing mode are a feature in which users can be differentiated. They used the degree of disturbance of the trigram typing time as a measure for the dissimilarity matrix, and also used a statistical method to calculate the average differences between the units in the matrix. This method is suitable for authentication because it requires predefined data.

All of the above techniques show that static authentication can be effectively used to distinguish users and to represent their typing behavior.

Continuous authentication uses the free text entered by each user for analysis, and this method is more accurate because it approaches real-world situations as compared to analyzing the static text entered by the user. In Table 2, we have described some of the most popular continuous authentication techniques.

Table 2. Continuous authentication methods.

Research	Number of Users	FAR (%)	FRR (%)	Accuracy (%)	Sample Content	Method
[19]	31	-	-	23	Some predefined and free sentences	Euclidean distance and probabilities
[18]	40	0	5.36	-	Two different texts, each 300 characters long	Distance measure
[20]	205	3.17	0.03	-	Each user typed 15 samples, and each sample contained between 700 and 900 characters	Nearest neighbor

In [19], an experiment was performed with 31 users, and data were collected for 7 weeks. The users had to type some predefined and some free sentences. This mechanism takes into account the characteristics that represent the user's behavior by calculating the average and standard deviation of the typing time of the digraphs, eliminating the residual values and thus determining the user's training profile, which is compared with the reference one through the use of Euclidean distance.

In [18], the time between the release of a key and the pressing of the next key was used for each pair of characters in a sequence. Forty users were invited to create a profile by entering two different texts. Ninety new users were invited to enter only the second text. The average distance was calculated between the unknown samples and each previously created profile (by the 40 users). The authors applied a learning scheme to improve the false rate and to calculate the mean and standard deviation between each sample in the user profile and each sample in the profile of another user. The results showed that FRR was reduced to 5.36% and FAR to zero.

In [20], the same technique was used as in [18] using the free text introduced by 205 users. The authors created user profiles based on the features of writing free text. Users performed a series of tests using entropy to measure the distance between the test sample and the reference samples from

any other user in the database. The examples are transformed into a list of n graphs sorted by their average times. In order to classify a sample, it is compared with each existing sample in both absolute and relative terms. This study is highly accurate when multiple users are registered.

All of the above techniques show that continuous authentication can be used, like static authentication, to effectively distinguish users. However, the characteristics of the users extracted by the continuous authentication method does not guarantee characteristics with strong statistical significance.

3. Algorithms and Methods

The algorithms used and described below are aimed at recognizing student users based on how they type and interact with the e-learning platform [21].

The first algorithm implemented refers to the recognition of a user's profile based on the behavior within the application [22]. Each user will have two profiles, one corresponding to the static text entered during authentication, and the second profile is created based on the activity within the platform during the session. Accordingly, the characteristics of this algorithm are measured from the user's first interaction with the application until its closure. The user's profile is determined based on how they use the keyboard through the following characteristics:

- typing speed;
- pressing time;
- how the user deletes text;
- how the user selects text;
- how the user chooses to write capitalized letters;
- the position of the control keys that the user uses.

The values for these characteristics are measured for each session of a user and saved in the database. For each user, a profile is created according to the algorithm from [22] and saved in the database.

Based on these values, the MED (Medium Euclidian Distance) set of the Euclidean distances between the point determined on the basis of the measurements made by the new user U and the profiles of all users in the database is formed:

$$MED = \{MED^1, MED^2, MED^3, \dots, MED^n\}$$

To identify a user U , it is sufficient to determine the minimum in the MED set. The user is identified by the user index corresponding to the minimum value in the MED set.

The second algorithm used refers to the static text entered by the user to be authenticated within the platform, i.e., how the user types the username and password. It measures the time of the first six digraphs in the username and password, thus obtaining 12 measurements for 12 groups. To be considered valid, each digraph must fall within 1.5 standard deviations of its mean. If 10 of 12 groups are significant (>83.33%) and also the condition settled by the first algorithm is fulfilled, the student is authenticated, and the values will be updated. In the following part, we describe the algorithm based on which the classification is performed:

Let U be the user trying to authenticate and gu, gp two sets of data that record the time of each group corresponding to the username (gu) and password (gp):

$$gu = \{gu_1, gu_2, gu_3, gu_4, gu_5, gu_6\}, gp = \{gp_1, gp_2, gp_3, gp_4, gp_5, gp_6\}$$

where gu_i $i = 1, 6$ is the value registered for group i in the username, and gp_i $i = 1, 6$ is the value registered for group i in the password.

The old values recorded for each group (values corresponding to the user U that is trying to authenticate) are taken from the database, in order to determine the confidence intervals in which the new values in the gu, gp sets must be found in order to be considered valid. Let k be a number

in the range 1, 6 and g_k be the k group in the username, where $g = \{g_1, g_2, g_3, g_4, g_5, g_6\}$ is the set of groups in the username. The set $X_{g_k}^U = \{x_1, x_2, x_3, x_4, x_5, x_6, \dots, x_n\}$ contains all the values previously recorded in the database for the group g_k , where n is the total number of user-registered authentications and x_1 is the value of the k group measured at the first authentication. In the same way, the data sets for the password are determined: $h = \{h_1, h_2, h_3, h_4, h_5, h_6\}$ is the set of groups and $Y_{h_k}^U = \{y_1, y_2, y_3, y_4, y_5, y_6, \dots, y_n\}$ is the set that contains all the values previously recorded in the database for the group $h_k, k = 1, 6$, where n is the total number of authentications.

For each group, $g_k, h_k, cu k = 1, 6$, is calculated:

- The average value of the group:

$$valMedie_{X_{g_k}^U} = \frac{\sum_{i=1}^n x_i}{n}, k = 1, 6 \tag{1}$$

where:

x_i —value i in the $X_{g_k}^U$ set;
 n —the number of registered authentications.

$$valMedie_{Y_{h_k}^U} = \frac{\sum_{i=1}^n y_i}{n}, k = 1, 6 \tag{2}$$

where:

y_i —value i in the $Y_{h_k}^U$ set.

- The standard deviation of each group:

$$std_{X_{g_k}^U} = \sqrt{\frac{\sum_{i=1}^n (x_i - valMedie_{X_{g_k}^U})^2}{n}}, k = 1, 6 \tag{3}$$

$$std_{Y_{h_k}^U} = \sqrt{\frac{\sum_{i=1}^n (y_i - valMedie_{Y_{h_k}^U})^2}{n}}, k = 1, 6 \tag{4}$$

- The confidence interval for each group:

$$valMedie_{X_{g_k}^U} - 1.5 * std_{X_{g_k}^U} \leq gu_k \leq valMedie_{X_{g_k}^U} + 1.5 * std_{X_{g_k}^U} \tag{5}$$

$$valMedie_{Y_{h_k}^U} - 1.5 * std_{Y_{h_k}^U} \leq gp_k \leq valMedie_{Y_{h_k}^U} + 1.5 * std_{Y_{h_k}^U} \tag{6}$$

where:

gu_k —the new value registered by the user for the k group in the username, $k = 1, 6$;
 1.5—value chosen experimentally to determine the length of the confidence interval;
 gp_k —the new user-registered value for group k in the password, $k = 1, 6$.

- The number of significant groups is determined (a group is significant if the new recorded value is within the confidence interval calculated based on the old values), and if this is over 10 (83.33% of values are significant) and the conditions settled by the first algorithm are also fulfilled, the student is authenticated; otherwise, they are rejected.

The third algorithm implemented for recognizing users based on how they type has the role of identifying students in a session when they enter any type of text. This method has the role of classifying students into valid or unauthorized users, especially after taking an exam/test/quiz. If an exam contains questions with free answers, at the end of the exam, the teacher is notified about the students who have been identified as unauthorized. Within this algorithm, the typing duration was

measured for the most common two-letter groups and also was performing the classification based on statistical methods as in the second algorithm described earlier. If during a test/quiz/exam, the student has used at least 4 of the 29 groups, the classification can be done. If the percentage of significance is over 83% and if the conditions of the first algorithm are met, the student has proved their authenticity, or on the contrary, if these conditions have not been met, the teacher is notified.

Let this be the set containing the 29 measured characteristics [14]:

$$C = \left\{ \begin{array}{l} C_{AL}, C_{AN}, C_{AR}, C_{AT}, C_{CH}, C_{CO}, C_{DI}, C_{EN}, C_{ER}, C_{ES}, C_{HE}, C_{IA}, C_{IN}, C_{IO}, C_{LAL}, \\ C_{NO}, C_{NT}, C_{ON}, C_{OR}, C_{PE}, C_{RA}, C_{RE}, C_{RI}, C_{ST}, C_{TA}, C_{TE}, C_{TI}, C_{TO}, C_{UN} \end{array} \right\}$$

where C is the general set.

An unknown student U^k cu $k = 1, m$ where m is the total number of students who have completed an exam following the next set of values:

$$CH^{U^k} = \left\{ \begin{array}{l} CH_{AL}^{U^k}, CH_{AN}^{U^k}, CH_{AR}^{U^k}, CH_{AT}^{U^k}, CH_{CH}^{U^k}, CH_{CO}^{U^k}, CH_{DI}^{U^k}, CH_{LA}^{U^k}, \\ CH_{EN}^{U^k}, CH_{ER}^{U^k}, CH_{ES}^{U^k}, CH_{HE}^{U^k}, CH_{IA}^{U^k}, CH_{IN}^{U^k}, CH_{IO}^{U^k}, CH_{NO}^{U^k}, \\ CH_{NT}^{U^k}, CH_{ON}^{U^k}, CH_{OR}^{U^k}, CH_{PE}^{U^k}, CH_{RA}^{U^k}, CH_{RE}^{U^k}, CH_{RI}^{U^k}, CH_{ST}^{U^k}, \\ CH_{TA}^{U^k}, CH_{TE}^{U^k}, CH_{TI}^{U^k}, CH_{TO}^{U^k}, CH_{UN}^{U^k} \end{array} \right\}$$

For each characteristic $C_i, i = 1, 29$, it checks to see if data have been recorded (if $CH_i^{U^k}$ is > 0) and is tested based on the statistical methods described in the second algorithm, whether that group was significant or not.

Let $D_{C_i}^{U^k} = \{d_1, d_2, \dots, d_n\}$, where $D_{C_i}^{U^k}$ is the set that contains all the values recorded by the user in database for the feature C_i , where n is the total number of non-zero values for that group.

- The average value of the group is calculated $D_{C_i}^{U^k}$ using the formula:

$$valMedie_{D_{C_i}^{U^k}} = \frac{\sum_{j=1}^n d_j}{n}, i = 1, 29 \tag{7}$$

where:

d_j —value from position j in the $D_{C_i}^{U^k}$ set;
 n —total number of values different from 0 for the feature C_i .

- The standard deviation of each group is determined using the formula:

$$std_{D_{C_i}^{U^k}} = \sqrt{\frac{\sum_{j=1}^n (d_j - valMedie_{D_{C_i}^{U^k}})^2}{n}}, i = 1, 29 \tag{8}$$

- The confidence intervals corresponding to each group are calculated, and if the new values are in the interval, then they are significant, otherwise they are not:

$$valMedie_{D_{C_i}^{U^k}} - 2 * std_{D_{C_i}^{U^k}} \leq CH_i^{U^k} \leq valMedie_{D_{C_i}^{U^k}} + 1.5 * std_{D_{C_i}^{U^k}} \tag{9}$$

where:

$CH_i^{U^k}$ —the new user-registered value for the group $i, i = 1, 29$;
 2—value chosen experimentally to determine the length of the confidence interval.

The number of significant groups is established, and if their percentage is over 83% then the student is identified as valid; otherwise, they are considered an unauthorized user.

4. Implementing Details for the Solution

Within this section, we describe how the values are determined based on which methods of user recognition are applied. The main used events are: *mousedown*, *keydown*, *keyup*, and also on all text-type inputs, the following properties are set to a false command: *copy*, *cut*, *paste*, *onDrag*, *onDrop*, and *autocomplete*, to force the student-type user to use the keyboard.

In the first algorithm, we mentioned that the following characteristics are measured: typing speed, pressing time, text selection mode, capitalization mode, text deletion mode, and the control keys used. Below is a detailed description of how measurements have been made for these characteristics:

- Typing speed is measured as the number of characters entered by the student in 6 s. Using a function that automatically starts refunctioning every 10 milliseconds, it is checked whether the student is typing, and as long as this condition is met, the start time is saved, and the number of characters entered is counted during the *keydown* event. Moreover, every 6 s another function is used, which performs the arithmetic mean between the number of characters typed previously and those typed in the current interval, resetting that counter for the new time interval to zero. The start time is retained for the situation where the student types for less than 6 s (or 5 s for authentication), and to determine how many characters were typed in x seconds, where x is a value < 6;
- Pressing time is the average value of pressing a key. The time of pressing a single key is calculated as the difference between the start time (saved on the *keydown* event) and that of the moment of release of the key (saved on the *keyup* event), and this value is saved in an array, which means that after stopping that activity, the average value will be determined;
- The way a student performs a deletion depends on the keys used for this mechanism, namely: Delete, Backspace, or mouse selection followed by pressing any key. Based on the code of the pressed keys (*keyCode*), a counter specific to each key is implemented, that is, if during the *keydown* event, the code of the key pressed is 8 then the counter specific to the Backspace key is increased. If it is 46, the one specific to the Delete key is increased. For deletion based on text selection, the *onSelect* event is treated on text-type inputs. If the text has been selected and the first event that is used is the *keydown*, then the counter specific for deletion by this method is incremented. Any event that is called after the *onSelect* resets the values. In the end, the three values are compared and the user's preference for text deletion is established;
- The control keys used, more precisely the preference for those positioned on the left side of the keyboard, or respectively, the right side, is determined using the *keydown* event. Specifically using a property of this event, namely, *event.location*, it returns 1 for the left keys, or respectively, 2 for the right hand keys. Depending on the returned result, it is checked if any of the shift, alt or ctrl keys have been pressed. If this condition is fulfilled, the specific counter for each key is increased, following which the values are finally compared and the way in which the student performs the operation or which control keys they use is established;
- Two events, namely, *onSelect* and *onKeyDown*, are used to determine how students prefer to select text. The *onSelect* event is triggered when the text is being selected, having a counter and increasing it every time this function is being used. The *keydown* event is checked (also based on the code of each key) if the shift key is pressed while the Left/Right keys are pressed. If the above condition is met, the specific counter is incremented, and at the end, these two values are compared and the way in which the user prefers to select the text is determined;
- Capitalization is done using Caps Lock, Shift, or added after typing. The sequence of keys that determines whether a user has used capital letters after writing the text is: *Left/Right/MouseDown* » *Backspace/Delete* » *Shift/Caps*. The *keydown* event is treated by comparing the *keyCode* of the key pressed with that of the keys mentioned above and increasing the value corresponding to each key. In the end, these values are compared and the user's preference for capitalization is determined.

All these values, described above, represent the input data for which the first algorithm described above applies.

The second implemented algorithm refers to the retention time of typing the first six groups of two letters in the username, and respectively, the duration of the first six groups of two letters in the password. To determine how long it takes, the *keydown* and *keyup* events are used, i.e., respectively, the pressing and releasing of a key. Two numeric variables are used, one of them showing which key in a group is pressed, and the other one indicating the group in which the recorded time is saved. If the first key in a group is pressed, the start time in a variable is saved on the *keydown* event, and when that value indicates the end of the group (it has the value 2), the difference between the actual time and the start time is determined on the *keyup* event, and this value is assigned to the group indicated by the second variable. An additional check is applied at the beginning of the *keydown* event in which it is checked whether the key pressed is alphanumeric, and if it is not, then it is ignored because it is not significant for this algorithm. Based on these values, the second method is applied. Moreover, in order to be able to make predictions about how a student is typing, at least four authentications are considered to be necessary so that the results provided by the algorithm are significant.

The third used algorithm analyzes how users type the most used 29 groups of two letters throughout the session. The events handled to determine durations are *keydown* and *keyup*. For each group, both its duration and the frequency of typing are retained. For example, for the word ERA, the duration of the ER and RA groups is recorded, which means that the *keyCode* of the first key in the group is retained on the *keydown* and compared with the *keyCode* of the second key on the *keyup* to determine if the two form a valid group (group from the predefined set). At the same time, the code of the second key is the first to be analyzed in the next group. On the *keydown* event corresponding to the first value in the group, the initial time is retained, and on the *keyup* event, the difference between the actual and the initial time is determined by adding it to the duration of the identified group. Its frequency is also increased. Finally, the average duration of the group is determined by dividing the total duration by the frequency of occurrence. In addition to these data, there is also a variable that shows whether or not these data come from taking an exam. If they do not come from an exam/quiz/test, the measured values are added to the database; otherwise, the third described algorithm is applied to identify the unauthorized users. In order for the predictions made on the basis of this algorithm to be true, there must be at least four exams/tests completed by the student in order to have reliable values and for the results provided to be as close to reality as possible.

5. Results

We have developed a learning management system whose main functionality is to identify users based on how they type, aimed at detecting unauthorized users to prevent system compromise and ensure data authenticity [23]. The students that will access the platform are entities, and they will have more sessions of interaction with the platform in order to create a profile based on their behavior [24]. There are two types of text entered by the user: static text and dynamic text. In this system, the analysis of the static text involves determining the profile of each user based on how he/she types the username and password to authenticate them within the platform, and if this profile corresponds to his/her previously created profile, the user will be allowed access within the application (analysis based on the first two algorithms described in the previous section). The analysis of dynamic text involves analyzing the text entered by the user during the session, when the user can introduce any type of text, aimed at determining the profiles of the students and classifying them into valid and unauthorized users after taking an exam (analysis based on the first and the last algorithm described in the previous section). The accuracy of these algorithms for student recognition is measured by two indicators: FAR and FRR. The methods of identifying students are based on statistical calculations and probabilities.

To test the system and determine these identifiers, we asked eight users to authenticate and take the exams and preparation tests within their courses to create a complete profile of each student (based on both the static and dynamic text typing during the session). Users had to take 10 exams and 4 tests,

and of these 14 supports, 4 were supported by an unauthorized user to determine the accuracy of the system during the session. In Tables 3 and 4 we present the results from tis experiment.

Table 3. The accuracy of authentication.

User	Number of Incorrect Acceptances of an Unauthorized User	Number of Incorrect Rejections of an Authorized User
<i>user1@securelms.com</i> Number of logins: 30	1	2
<i>user2@securelms.com</i> Number of logins: 20	0	2
<i>user3@securelms.com</i> Number of logins: 14	2	2
<i>user4@securelms.com</i> Number of logins: 14	1	1
<i>user5@securelms.com</i> Number of logins: 14	0	0
<i>user6@securelms.com</i> Number of logins: 14	4	0
<i>user7@securelms.com</i> Number of logins: 14	2	1
<i>user8@securelms.com</i> Number of logins: 14	1	1
Total number of authentications: 134		

Table 4. The accuracy of continuous techniques.

User	Number of Incorrect Acceptances of an Unauthorized User	Number of Incorrect Rejections of an Authorized User
<i>user1@securelms.com</i> Number of sessions: 14	0	1
<i>user2@securelms.com</i> Number of sessions: 14	0	0
<i>user3@securelms.com</i> Number of sessions: 14	1	2
<i>user4@securelms.com</i> Number of sessions: 14	1	0
<i>user5@securelms.com</i> Number of sessions: 14	0	2
<i>user6@securelms.com</i> Number of sessions: 14	1	1
<i>user7@securelms.com</i> Number of sessions: 14	0	2
<i>user8@securelms.com</i> Number of sessions: 14	1	2
Total number of sessions: 112		

Based on the values identified in the Table 3, the two indicators are calculated as follows:

$$FAR_{login} = \frac{\text{Number of incorrect acceptances of an unauthorized user}}{\text{total number of authentications}} = \frac{11}{134} * 100 = 8.2\% \quad (10)$$

$$FRR_{login} = \frac{\text{Number of incorrect rejections of an authorized user}}{\text{total number of authentications}} = \frac{9}{134} * 100 = 6.71\% \quad (11)$$

As we can see, for a number of 134 authentications, we obtained an FAR of 8.2% and an FRR of 6.71%.

Based on the values, from Table 4, obtained during the session, we can determine the two indicators as follows:

$$FAR_{session} = \frac{\text{Number of incorrect acceptances of an unauthorized user}}{\text{Total number of sessions}} = \frac{4}{112} * 100 = 3.57\% \quad (12)$$

$$FRR_{session} = \frac{\text{Number of incorrect rejections of an authorized user}}{\text{Total number of sessions}} = \frac{10}{112} * 100 = 8.92\% \quad (13)$$

The results during a working session show that the FAR is 3.57% and the FRR is 8.92%. The obtained results for the FAR are comparable with the results of previous research studies, but the obtained results for FRR are weaker than the results obtained in previous research due to the number of sessions and due to the used algorithm, which implements more techniques than previous research studies. These values can be improved by increasing the number of sessions for each user until the evaluation. In these sessions, the algorithm will create the profile of the user, and this profile will be more accurate if the number of sessions is greater.

Within the system, users identified as unauthorized during the exam are presented in both the teacher's general report and the one corresponding to the individual exam, as shown in Figure 2.

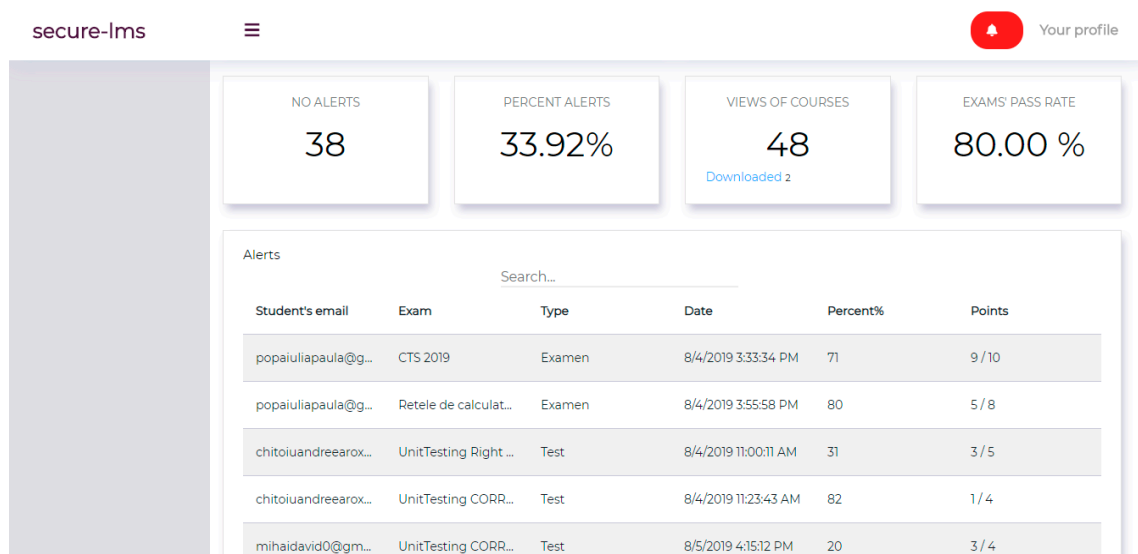


Figure 2. Unauthorized users identified.

As can be seen in Figure 2, the total number of unauthorized users is 38. As mentioned above, 4 of the 14 exams for each student were taken by an unauthorized user. In the most favorable case, for FAR and FRR to be 0% (number of incorrect acceptances of an unauthorized user = 0; number of incorrect rejections of an authorized user = 0), the number of alerts identified would have been 32.

The results of these methods for identifying students based on statistical calculations and probabilities would become more accurate if there were more authentications performed and more exams taken, which would also involve a decrease in FRR and FAR indicators over time.

6. Conclusions

User detection based on the dynamics of real-time typing is a biometric mechanism that prevents improper authentication and false identity. This paper focuses on the development of automatic user

recognition techniques throughout users' interaction with the system, aiming to reduce computer vulnerabilities by detecting intruders and preventing the system from being compromised, so as to ensure the authenticity of the data. The main motivation for the development of this system was the need for a flexible system to replace the standard learning methods, but at the same time to ensure the integrity of data. This motivation also comes from the lack of an automatic continuous authentication system based on the dynamics of the keyboard, a system that does not require additional hardware equipment, therefore being suitable as the users use the keyboard at least for authentication.

This system designed for higher education could help to improve students' academic performance because it integrates two essential aspects of the learning process, namely, collaboration and motivation [25]. At the moment, students' demand for the digitalization of the learning process is constantly increasing due to competitiveness, as exams generate feedback allowing the students to self-improve by stimulating their need for self-realization [26]. At the same time, this system allows for the integration of the most important components of the learning process, namely, accumulating knowledge through courses and testing it through verification tests.

This research has been conducted to verify the existing algorithms and the new proposed algorithm to recognize the user on the learning management system. The results were obtained by conducting a test on our implemented platform with these algorithms. The results show that students can be recognized only if they have an intense activity on the platform until the evaluation. This activity is necessary to create the user profile in the database.

In our future research, we want to develop more options for this system. One possibility for developing it would be to include practical programming exams. The application would behave like an integrated development environment (IDE) that would compile real-time code written by the student in the chosen programming language. The identification of students would be based on a model that contains groups of characters and symbols, depending on the particularities of the chosen programming language. This functionality would facilitate a student's experience in a learning management system while ensuring the authenticity of the data.

Author Contributions: A.Z.: overall conceptualization, investigation, methodology, resources, software, writing—original draft preparation; D.C.: investigation, methodology, resources, software development, experiments, analysis of results; M.Z. and C.T.: methodology, introduction and conclusions, writing—review and editing. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Araújo, L.C.; Sucupira, L.H.; Lizarraga, M.G.; Ling, L.L.; Yabu-Uti, J.B.T. User authentication through typing biometrics features. *IEEE Trans. Signal Process.* **2005**, *53*, 851–855. [[CrossRef](#)]
2. Jain, A.; Ross, A.; Prabhakar, S. An introduction to biometric recognition. *IEEE Trans. CSVT* **2004**, *14*, 20. [[CrossRef](#)]
3. Vacca, J.R. *Biometric Technologies and Verification Systems*; Elsevier: Amsterdam, The Netherlands, 2007; p. 607.
4. Barros, A.; Resque, P.; Almeida, J.; Mota, R.; Oliveira, H.; Rosário, D.; Cerqueira, E. Data Improvement Model Based on ECG Biometric for User Authentication and Identification. *Sensors* **2020**, *20*, 2920. [[CrossRef](#)] [[PubMed](#)]
5. Filiz, D.; Tanrıöver, Ö.Ö. An Exploration of Machine Learning Methods for Biometric Identification Based on Keystroke Dynamics. In *Artificial Intelligence and Machine Learning Applications in Civil, Mechanical, and Industrial Engineering*; IGI Global: Hershey, PA, USA, 2020; pp. 258–269.
6. Raul, N.; Shankarmani, R.; Joshi, P. A Comprehensive Review of Keystroke Dynamics-Based Authentication Mechanism. In *International Conference on Innovative Computing and Communications*; Springer: Singapore, 2020; pp. 149–162.

7. Mustafic, T.; Camtepe, S.; Albayrak, S. Continuous and non-intrusive identity verification in real-time environments based on free-text keystroke dynamics. In Proceedings of the 2011 International Joint Conference on Biometrics (IJCB), Washington, DC, USA, 11–13 October 2011.
8. Banerjee, S.P.; Woodard, D. Biometric authentication and identification using keystroke dynamics: A survey. *J. Pattern Recognit. Res.* **2012**, *7*, 116–139. [[CrossRef](#)]
9. Joseph, R.; Liu, X.; Metaxas, D. On Continuous User Authentication via Typing Behavior. *IEEE Trans. Image Process.* **2014**, *28*, 4611–4624.
10. Manish, S. Password Authentication Method Using Keystroke Biometric. *J. Comput.* **2011**, *3*, 125–129.
11. Solano, J.; Camacho, L.; Correa, A.; Deiro, C.; Vargas, J.; Ochoa, M. Risk-based static authentication in web applications with behavioral biometrics and session context analytics. In *International Conference on Applied Cryptography and Network Security, Proceedings of the Applied Cryptography and Network Security Workshops, Bogota, Colombia, 5–7 June 2019*; Zhou, J., Deng, R., Li, Z., Majumdar, S., Meng, W., Wang, L., Zhang, K., Eds.; Springer: Cham, Switzerland, 2019; pp. 3–23.
12. Shi, E.; Niu, Y.; Jakobsson, M.; Chow, R. *Implicit Authentication through Learning User Behavior*; Information Security; Springer: Berlin/Heidelberg, Germany, 2011; pp. 99–113.
13. Yampolskiy, R.V. Action-based User Authentication. *Int. J. Electron. Secur. Digit. Forensics* **2008**, *1*, 3. [[CrossRef](#)]
14. Alsolami, E. An examination of keystroke dynamics for continuous user authentication. Ph.D. Thesis, Queensland University of Technology, Brisbane, Australia, 2012.
15. Gaines, R.S.; Lisowski, W.; Press, S.J.; Shapiro, N. *Authentication by Keystroke timing: Some Preliminary Results (No. RAND-R-2526-NSF)*; Rand Corp: Santa Monica, CA, USA, 1980.
16. Umphress, D.; Williams, G. Identity verification through keyboard characteristics. *Int. J. Man Mach. Stud.* **1985**, *23*, 263–273. [[CrossRef](#)]
17. Obaidat, M.S.; Sadoun, B. Verification of computer users using keystroke dynamics. *IEEE Trans. Syst. Man Cybern. Part B* **1997**, *27*, 261–269. [[CrossRef](#)] [[PubMed](#)]
18. Bergadano, F.; Gunetti, D.; Picardi, C. Identity verification through dynamic keystroke analysis. *Intell. Data Anal.* **2003**, *7*, 469–496. [[CrossRef](#)]
19. Monroe, F.; Rubin, A. Authentication via keystroke dynamics. In Proceedings of the 4th ACM Conference on Computer and Communications Security, Zurich, Switzerland, 1–4 April 1997; pp. 48–56.
20. Bergadano, F.; Gunetti, D.; Picardi, C. User authentication through keystroke dynamics. *ACM Trans. Inf. Syst. Secur.* **2002**, *5*, 367–397. [[CrossRef](#)]
21. Jagadamaba, G. Keystroke Dynamics in E-Learning and Online Exams. In *Biometric Authentication in Online Learning Environments*; IGI Global: Hershey, PA, USA, 2019; pp. 1–21.
22. Zamfiroiu, A.; Ciurea, C. A Model for Users 'profile Recognition Based on Their Behavior in Online Applications. *Econ. Comput. Econ. Cybern. Stud. Res.* **2017**, *2*, 181–194.
23. Boja, C.; Doinea, M.; Pocatilu, P. Impact of the security requirements on mobile applications usability. *Econ. Inform.* **2013**, *13*, 64.
24. Uta, A.; Ivan, I.; Popa, M.; Ciurea, C.; Doinea, M. Security of virtual entities. In Proceedings of the 15th International Conference on Computer Systems and Technologies, Ruse, Bulgaria, 27–28 June 2014; pp. 278–285.
25. Iancu, B. Gamification Applied in Computer Science Education: A Preliminary Approach. *Econ. Inform.* **2019**, *19*, 52–58.
26. Ciobanu, R.C. *Implementation of Mobile Solutions in Romania's Education System*; Informatica Economica: Bucharest, Romania, 2019; p. 23.

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



© 2020 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).